CSE 535 Mobile Computing - Project 4 SmartLife Companion

Ruchitha Mekala
1225704532

Abstract

The "SmartLife Companion" (Guardian Angel) initiative is focused on developing a versatile app ecosystem designed to effortlessly blend into the daily routines of users, offering a variety of tailored services. Essential functions of this system encompass health monitoring and preliminary diagnosis, student support, fall detection, and mood management. As a part of project 4, I have implemented the fall detection algorithm using accelerometer triaxial data.

Alignment with guardian angel

Fall detection in our project - “Smart Companion” adds a critical layer of safety by detecting falls, which is particularly beneficial for elderly users or individuals with health conditions prone to falls. Works in conjunction with health data tracking, providing a holistic safety approach where a fall could be correlated with health anomalies detected by the app. Although the primary objective of the feature is personal safety, there could also be an extra layer of utility of the feature, where the phone falls from the pocket without the user’s notice. The alarm can alert the user for potential loss of the phone.

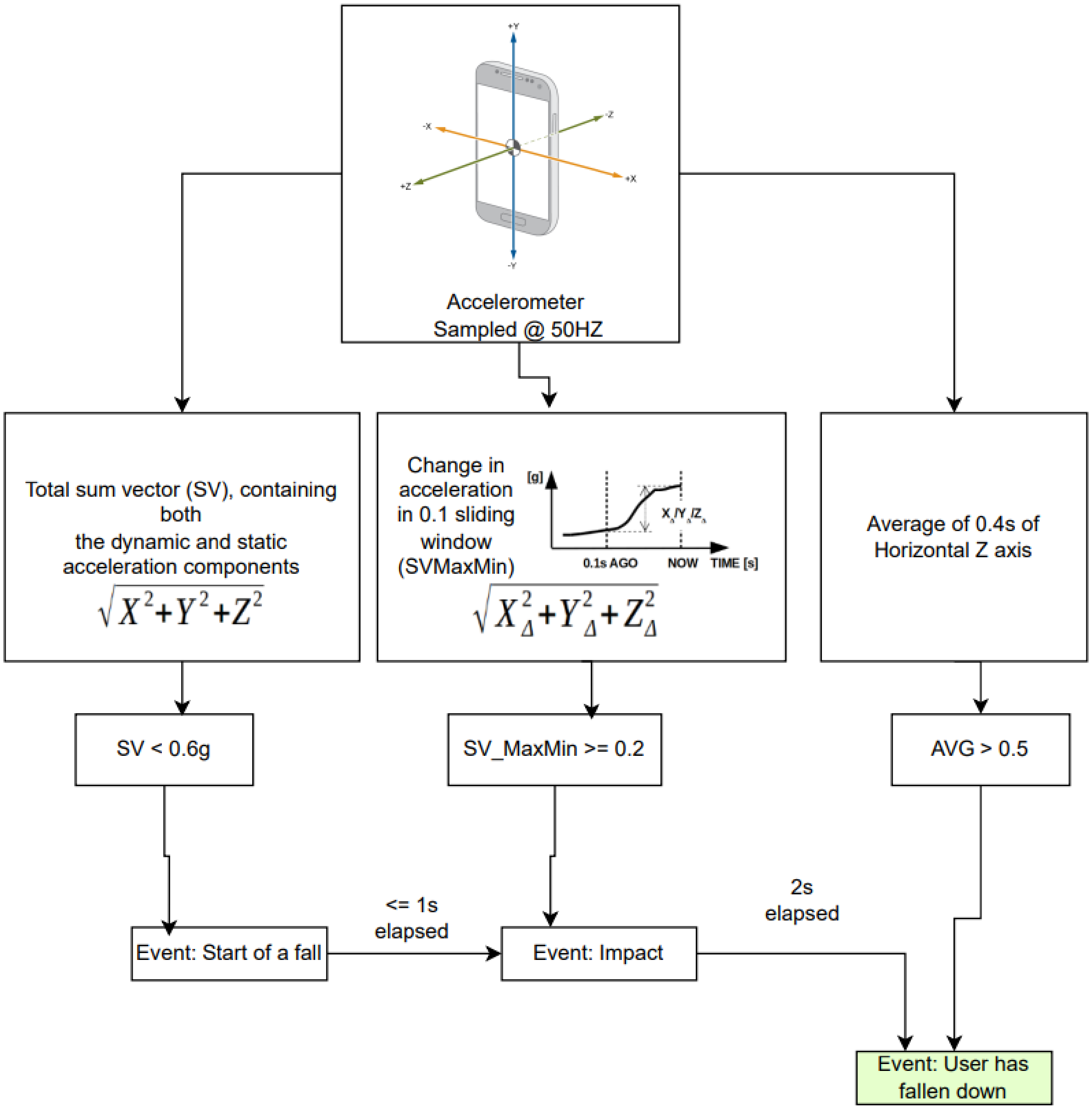
Specifications -

- The feature runs in the background and keeps collecting accelerometer triaxial data sampled at 50HZ for computation.
- Sophisticated parameters have been formed using x, y and z axis values.
  - Parameter 1 - Start of the fall = $\sqrt{X^2 + Y^2 + Z^2}$
  - Parameter 2 - Impact = $(X_\Delta^2 + Y_\Delta^2 + Z_\Delta^2)^{0.5}$
  - Parameter 3 - Lying after the fall = Average of z axis value for the last 0.4 seconds

- Start of the fall contains both the dynamic and static acceleration components,
- Impact is the change in the acceleration signal that was investigated by constructing a sliding sum vector SVMaxMin, which was calculated using the differences between the maximum and minimum values in a 0.1 s sliding window for each axis.

- Thresholds have been set to each parameter, and on crossing each threshold, the event would be flagged as a ‘Fall’. The thresholds have been fixed upon referring from ‘Comparison of low-complexity fall detection algorithms for body attached accelerometers’ paper published in Elsevier.

Design


+Y
-Z
-X
+X
+Z
-Y
Accelerometer Sampled @ 50HZ
Total sum vector (SV), containing both the dynamic and static acceleration components
$\sqrt{X^2+Y^2+Z^2}$
Change in acceleration in 0.1 sliding window (SVMaxMin)
[g]
$X_\Delta/Y_\Delta/Z_\Delta$
0.1s AGO
NOW
TIME [s]
$\sqrt{X_\Delta^2+Y_\Delta^2+Z_\Delta^2}$
Average of 0.4s of Horizontal Z axis
SV < 0.6g
SV_MaxMin >= 0.2
AVG > 0.5
Event: Start of a fall
<= 1s elapsed
Event: Impact
2s elapsed
Event: User has fallen down

Testing Strategies

Rather than setting separate thresholds for the x, y, and z values, advanced parameters (as previously described) are utilized to distinguish between a phone being accidentally dropped and an actual fall by a person. Live environment testing was conducted by installing the APK on an Android phone. Additionally, the height from which the phone falls was factored into the algorithm, which has been calibrated to recognize falls from an average height.

Navigation Challenges

Managing and navigating the flow of alerts and notifications, particularly for false alarms versus actual falls, required meticulous attention. Differentiating these alerts and ensuring they reached the user without overwhelming them was a complex task.